A rank-exploiting infinite Arnoldi algorithm for nonlinear eigenvalue problems

Roel Van Beeumen
Elias Jarlebring
Wim Michiels

Report TW 652, July 2014

KU Leuven
Department of Computer Science
Celestijnenlaan 200A – B-3001 Heverlee (Belgium)

A rank-exploiting infinite Arnoldi algorithm for nonlinear eigenvalue problems

Roel Van Beeumen
Elias Jarlebring
Wim Michiels

Report TW 652, July 2014

Department of Computer Science, KU Leuven

Abstract

We consider the nonlinear eigenvalue problem: $M(\lambda)x = 0$, where $M(\lambda)$ is a large parameter-dependent matrix. In several applications, $M(\lambda)$ has a structure where the higher-order terms of its Taylor expansion have a particular low-rank structure. We propose a new Arnoldi based algorithm that can exploit this structure. More precisely, the proposed algorithm is equivalent to Arnoldi's method applied to an operator whose reciprocal eigenvalues are solutions to the nonlinear eigenvalue problem. The iterates in the algorithm are functions represented in a particular structured vector-valued polynomial basis similar to the construction in the infinite Arnoldi method [Jarlebring, Michiels, and Meerbergen, Numer. Math., 122 (2012), pp. 169–195]. In this paper the low-rank structure is exploited by applying an additional operator and by using a more compact representation of the functions. This reduces the computational cost associated with orthogonalization, as well as the required memory resources. The structure exploitation also provides a natural way carrying out implicit restarting and locking without the need to impose structure in every restart. The efficiency and properties of the algorithm are illustrated with two large-scale problems.

Keywords : nonlinear eigenvalue problem, Arnoldi method, low-rank
MSC : 47J10, 65F15.

A rank-exploiting infinite Arnoldi algorithm for nonlinear eigenvalue problems

Roel Van Beeumen[†] Elias Jarlebring[‡] Wim Michiels[†]

July 21, 2014

Abstract

We consider the nonlinear eigenvalue problem: $M(\lambda)x = 0$, where $M(\lambda)$ is a large parameter-dependent matrix. In several applications, $M(\lambda)$ has a structure where the higher-order terms of its Taylor expansion have a particular low-rank structure. We propose a new Arnoldi based algorithm that can exploit this structure. More precisely, the proposed algorithm is equivalent to Arnoldi's method applied to an operator whose reciprocal eigenvalues are solutions to the nonlinear eigenvalue problem. The iterates in the algorithm are functions represented in a particular structured vector-valued polynomial basis similar to the construction in the infinite Arnoldi method [Jarlebring, Michiels, and Meerbergen, Numer. Math., 122 (2012), pp. 169–195]. In this paper the low-rank structure is exploited by applying an additional operator and by using a more compact representation of the functions. This reduces the computational cost associated with orthogonalization, as well as the required memory resources. The structure exploitation also provides a natural way carrying out implicit restarting and locking without the need to impose structure in every restart. The efficiency and properties of the algorithm are illustrated with two large-scale problems.

Keywords: nonlinear eigenvalue problem; Arnoldi method; low-rank

1 Introduction

Suppose $\Omega \subset \mathbb{C}$ is a closed disk centered at the origin and let $M : \Omega \to \mathbb{C}^{n\times n}$ be a matrix with elements which are analytic in Ω. We will consider the problem of finding $(\lambda, x) \in \Omega \times \mathbb{C}^n \backslash \{0\}$ such that

$$M(\lambda)x = 0. \tag{1.1}$$

This nonlinear eigenvalue problem occurs in many situations. For instance, they arise in the study of stability of higher-order differential equations where they give rise to quadratic and polynomial eigenvalue problems [2, 24]; in the study of delay-differential equations [12]; and in the study of fluid-solid interaction where $M(\lambda)$ contains rational functions [27]. There are also problems involving boundary integral operators [25]. For summary works and benchmark collections on nonlinear eigenvalue problems we refer to [4, 29, 18].

There are many algorithms in various generality settings for solving nonlinear eigenvalue problems, e.g., based on Arnoldi's method [27], Jacobi-Davidson methods [28, 21],

[†]Department of Computer Science, KU Leuven, University of Leuven, 3001 Heverlee, Belgium (Roel.VanBeeumen@cs.kuleuven.be, Wim.Michiels@cs.kuleuven.be).

[‡]Department of Mathematics, NA group, KTH Royal Institute of Technology, 100 44 Stockholm, Sweden (eliasj@kth.se).

methods which can be seen as flavors and extensions of Newton's method [19, 16, 7], and contour integral formulations [1, 5]. There are also several approaches based on first approximating $M(\lambda)$ and subsequently linearizing the approximation. This gives rise to companion-type linearizations from which the structure can be exploited [8, 26, 11]. However, our approach here is similar to [14] and based on directly applying the Arnoldi method on an operator reformulation of the nonlinear eigenvalue problem (1.1).

In the recent literature, there are several nonlinear eigenvalue methods that exploit low-rank structures in different settings, e.g., the approach for low-rank modifications of symmetric eigenvalue problems in [30] and the linearization approach for rational eigenvalue problems in [23] and for nonlinear eigenvalue problems in [26, 11]. In this paper we will exploit a particular type of low-rank structure combined with the infinite Arnoldi method [14]. This method is equivalent to Arnoldi's method applied to an operator and the restarting procedure is based on the structure of the invariant subspace presented in [13].

In this work we will construct an algorithm which exploits a particular commonly occuring structure, i.e., the high-order coefficients in the Taylor expansion of M have low rank. More precisely, let M have a Taylor expansion denoted by

$$M(\lambda) = M_0 + \frac{\lambda}{1!}M_1 + \frac{\lambda^2}{2!}M_2 + \cdots, \tag{1.2}$$

and assume that the higher-order terms have low-rank structure in the sense that

$$M_i = V_i Q^*, \qquad i \geq p+1, \tag{1.3}$$

where $Q \in \mathbb{C}^{n\times r}$ is a matrix with orthonormal columns and $V_i \in \mathbb{C}^{n\times r}$ for $i \geq p+1$. The proposed algorithm will be particularly suitable for large n and situations where $r \ll n$. This low-rank property often appears in the discretizations of PDE eigenvalue problems that have been constructed with non-reflecting boundary conditions. In numerical examples we will also give an example of how localized delayed feedback control can give rise to this type of low-rank structure.

The paper is organized as follows. In Section 2, we will show that the solutions to (1.1) are reciprocal eigenvalues of an operator $\mathcal{FB}$. This is similar to [14] where an equivalence was shown for $\mathcal{B}$. The additional operator $\mathcal{F}$ stems from the low-rank structure and allows for considerable perfomance improvement. In Section 3, we consider the Arnoldi method for the operator $\mathcal{FB}$ where we represent the iterates in a polynomial basis. We also show that, if we start the iteration in a particular way and use a particular vector-valued polynomial basis, we can carry out the Arnoldi method for $\mathcal{FB}$ very efficiently. In comparison to [14] the basis matrix in this algorithm grows slower yielding a reduction of the computation time required for the orthogonalization and the memory resources required to store the basis matrix. The slower growth also allows for a natural way to carry out implicit restarting and locking. This is derived in Section 4. It is well know that the Arnoldi's method usually converges quickly to extreme isolated eigenvalues, see e.g. [20, Section 6.7]. As a consequence of the fact that we carry out the Arnoldi method on an operator with inverted eigenvalue set, the construction is likely to find solutions to (1.1) close to the origin quickly, similar to shift-and-invert Arnoldi method. This as well as other efficiency properties are illustrated in the numerical experiments in Section 5.

2 Equivalent linear operator eigenvalue problem

Similarly as in the infinite Arnoldi method [14], the basis of the algorithm will be a characterization of the solutions to (1.1) as reciprocal eigenvalues of a linear operator. Here, we will use an operator that also takes the low-rank structure into account.

If we let

$$B(\lambda) := \frac{1}{\lambda} M(0)^{-1}(M(0) - M(\lambda)),$$

we have that

$$\lambda B(\lambda)x = x, \qquad \lambda \in \mathbb{C}, \quad x \in \mathbb{C}^n \backslash \{0\}, \tag{2.1}$$

unless $\lambda = 0$, and define $B(0)$ with analytic continuation. If $M(\lambda)$ satisfies (1.3), then $B(\lambda)$ will satisfy a similar low-rank property. Therefore, we decompose $B(\lambda)$ as follows

$$B(\lambda) = B_{\text{pol}}(\lambda) + B_{\text{rem}}(\lambda),$$

where the highest power in B_{pol} is λ^{p-1} (whereas the highest full rank power in M is λ^p). We have more precisely the following:

- B_{pol} is a polynomial matrix of a given degree $p-1$ corresponding to the first p terms in the Taylor series of B, i.e., it can be expanded as

$$B_{\text{pol}}(\lambda) = B_0 + B_1 \frac{\lambda}{1!} + B_2 \frac{\lambda^2}{2!} + \cdots + B_{p-1} \frac{\lambda^{p-1}}{(p-1)!}.$$

- B_{rem} is the remainder of the Taylor expansion of B and can be expanded as

$$B_{\text{rem}}(\lambda) = B_p \frac{\lambda^p}{p!} + B_{p+1} \frac{\lambda^{p+1}}{(p+1)!} + \cdots .$$

Note that B_{rem} is of row rank in the sense that there exists a matrix $Q \in \mathbb{C}^{n \times r}$, with r possibly small, such that the following factorization holds:

$$B_i = U_i Q^*, \qquad i \geq p, \tag{2.2}$$

where $U_i = \frac{1}{i+1} M(0)^{-1} V_{i+1}$.

The construction of the algorithm requires two operators:

- The operator $\mathcal{B} : \mathcal{C}(\mathbb{R}, \mathbb{C}^n) \to \mathcal{C}(\mathbb{R}, \mathbb{C}^n)$, which served as a basis of the derivation in [14], is defined by

$$(\mathcal{B}\,\phi)(\theta) = \int_0^\theta \phi(s) ds + \left(B\left(\frac{d}{d\theta}\right) \phi \right)(0), \tag{2.3}$$

  where the integration constant is given by

$$\left(B\left(\frac{d}{d\theta}\right) \phi \right)(0) = \sum_{i=0}^{\infty} \frac{1}{i!} B_i \phi^{(i)}(0).$$

- The operator $\mathcal{F} : \mathcal{C}(\mathbb{R}, \mathbb{C}^n) \to \mathcal{C}(\mathbb{R}, \mathbb{C}^n)$ is defined by

$$(\mathcal{F}\phi)(\theta) = \sum_{i=0}^{p-1} \phi^{(i)}(0) \frac{\theta^i}{i!} + \sum_{i=p}^{\infty} Q Q^* \phi^{(i)}(0) \frac{\theta^i}{i!},$$

We will now relate (2.1) with the operator eigenvalue problem

$$\lambda(\mathcal{F}\mathcal{B})\phi = \phi, \qquad \mathcal{F}\mathcal{B} \in \mathcal{C}^\infty(\mathbb{R}, \mathbb{C}^n), \quad \lambda \in \mathbb{C}, \quad \phi \not\equiv 0. \tag{2.4}$$

The eigenvalue problems (2.1) and (2.4) are equivalent in the following sense.

Theorem 2.1 (Operator equivalence)**.** *Suppose $B : \Omega \to \mathbb{C}^{n\times n}$ has matrix elements analytic in Ω and suppose $B(\lambda)$ satisfies the low-rank property* (2.2)*. Then, the following implications are satisfied.*

1. *Let the pair (λ, x), with $\lambda \neq 0$, be a solution of (2.1). Then (λ, ϕ) is a solution of* (2.4)*, where*

$$\phi(\theta) = \sum_{i=0}^{p-1} \frac{(\lambda\theta)^i}{i!} x + \sum_{i=p}^{\infty} \frac{(\lambda\theta)^i}{i!} QQ^* x. \tag{2.5}$$

2. *Let the pair (λ, ϕ), with $\lambda \neq 0$, be a solution of (2.4). Then (λ, x), with $x = \phi(0)$, is a solution of* (2.1)*.*

Proof. The proof consists of two parts.
Statement 1. Let ϕ be given by (2.5). We have that

$$\left(B_{\mathrm{pol}}\left(\frac{d}{d\theta}\right)\phi\right)(0) = B_{\mathrm{pol}}(\lambda)x \tag{2.6}$$

$$\left(B_{\mathrm{rem}}\left(\frac{d}{d\theta}\right)\phi\right)(0) = B_{\mathrm{rem}}(\lambda)QQ^*x \tag{2.7}$$

and from the definition of $\mathcal{B}$, it follows that

$$(\mathcal{B}\,\phi)(\theta) = \sum_{i=0}^{p-1} \frac{\lambda^i\theta^{i+1}}{(i+1)!} x + \sum_{i=p}^{\infty} \frac{\lambda^i\theta^{i+1}}{(i+1)!} QQ^*x + B_{\mathrm{pol}}(\lambda)x + B_{\mathrm{rem}}(\lambda)QQ^*x.$$

Consequently

$$((\mathcal{F}\mathcal{B})\,\phi)(\theta) = \sum_{i=1}^{p-1} \frac{\lambda^{i-1}\theta^{i}}{i!} x + \sum_{i=p}^{\infty} \frac{\lambda^{i-1}\theta^{i}}{i!} QQ^*x + B_{\mathrm{pol}}(\lambda)x + B_{\mathrm{rem}}(\lambda)QQ^*x.$$

From (2.2) and the fact that (λ, x) is an eigenpair we get

$$B_{\mathrm{pol}}(\lambda)x + B_{\mathrm{rem}}(\lambda)QQ^*x = B_{\mathrm{pol}}(\lambda)x + B_{\mathrm{rem}}(\lambda)x = B(\lambda)x = \frac{1}{\lambda}x,$$

and it follows that

$$\lambda(\mathcal{F}\mathcal{B})\phi = \phi.$$

Statement 2. Let (λ, ϕ) be an eigenpair of $\mathcal{F}\mathcal{B}$ and denote the power series expansion of ϕ by

$$\phi = \sum_{i=0}^{\infty} \theta^i x_i.$$

Hence, we get

$$\mathcal{B}\phi = \left(B\left(\frac{d}{d\theta}\right)\phi\right)(0) + \sum_{i=0}^{\infty} \frac{\theta^{i+1}}{i+1} x_i,$$

$$(\mathcal{F}\mathcal{B})\,\phi = \left(B\left(\frac{d}{d\theta}\right)\phi\right)(0) + \sum_{i=0}^{p-1} \frac{\theta^{i+1}}{i+1} x_i + \sum_{i=p}^{\infty} \frac{\theta^{i+1}}{i+1} QQ^* x_i.$$

Equating the coefficients corresponding to powers of θ in $(\mathcal{FB})\phi = \frac{1}{\lambda}\phi$ yields:

$$
\begin{aligned}
B\left(\frac{d}{d\theta}\right)\phi(0) &= \frac{1}{\lambda}x_0 \\
\frac{x_0}{1} &= \frac{1}{\lambda}x_1 \\
&\vdots \\
\frac{x_p}{p-1} &= \frac{1}{\lambda}x_{p-1} \\
QQ^*\frac{x_{p-1}}{p} &= \frac{1}{\lambda}x_p \\
QQ^*\frac{x_p}{p+1} &= \frac{1}{\lambda}x_{p+1} \\
&\vdots
\end{aligned}
\tag{2.8}
$$

We conclude directly that

$$
x_i = \begin{cases} \frac{\lambda^i}{i!}x_0, & i = 1, \ldots, p-1, \\ \frac{\lambda^i}{i!}QQ^*x_0, & i \geq p. \end{cases}
$$

Since by assumption $\phi \not\equiv 0$, we see that $x_0 \neq 0$ must hold. Writing out the first equation of (2.8) yields

$$
x_0 = \lambda \sum_{i=0}^{\infty} B_i x_i = \lambda B_{\text{pol}}(\lambda)x_0 + \lambda B_{\text{rem}}(\lambda)QQ^*x_0 = \lambda B(\lambda)x_0.
$$

We conclude that (λ, x_0) is an eigenpair of (2.1). The proof is completed by noticing that $x_0 = \phi(0)$. □

3 Arnoldi's method on $\mathcal{FB}$

The infinite Arnoldi method [14] is equivalent to Arnoldi's method for the operator $\mathcal{B}$ whose reciprocal eigenvalues are solutions to (2.1). We know from Theorem 2.1 that the reciprocal eigenvalues of the operator $\mathcal{FB}$ are also solutions to (2.1) for problems with low-rank structure. Analogous to the infinite Arnoldi method, we will now construct an algorithm by considering Arnoldi's method for $\mathcal{FB}$. By carrying out k steps of Arnoldi's method for $\mathcal{FB}$ with a starting function ϕ, we generate a sequence of functions $\phi_1, \ldots, \phi_k$ forming a basis of the Krylov subspace

$$
\begin{aligned}
\mathcal{K}_k(\mathcal{B}, \phi) &:= \operatorname{span}\left\{\phi,\ \mathcal{FB}\phi,\ \ldots,\ (\mathcal{FB})^{k-1}\phi\right\}, \\
&:= \operatorname{span}\left\{\phi_1,\ \phi_2,\ \ldots,\ \phi_k\right\}.
\end{aligned}
\tag{3.1}
$$

If we start the Arnoldi method with a constant function, we can show from the fact that $\mathcal{FB}$ corresponds to integration, that $\phi_1, \ldots, \phi_k$ are vector-valued polynomials. However, unlike [14], we will here see that due to the application of $\mathcal{F}$, the functions can be represented with less information yielding a significant performance improvement. This will also allow to carry out implicit restarting in a natural way which we shall explain in Section 4. The possibility to represent the functions in a compressed way stems from the fact that some polynomial coefficients can be represented with vectors of smaller size. This can be seen from the following lemma, where we see that the polynomial coefficients of degree higher than $p-1$ can be represented with vectors of length r, i.e., the rank of the low-rank terms.

Lemma 3.1. *Suppose $\phi_1(\theta) := z_0$ is constant. Then there are constants $x_0, \ldots, x_{p-1} \in \mathbb{C}^n$ and $\hat{x}_p, \ldots, \hat{x}_k \in \mathbb{C}^r$ such that*

$$((\mathcal{F}\mathcal{B})^k \phi_1)(\theta) = x_0 + x_1\theta + x_2\theta^2 + \cdots + x_{p-1}\theta^{p-1} + Q(\hat{x}_p\theta^p + \cdots + \hat{x}_p\theta^k). \tag{3.2}$$

Proof. The result follows directly from the definitions of $\mathcal{B}$ and $\mathcal{F}$. □

It is also easy to show that Arnoldi's method applied to $\mathcal{F}\mathcal{B}$, i.e., [14, Algorithm 1] with $\mathcal{F}\mathcal{B}$ instead of $\mathcal{B}$, generates iterates $\phi_1, \ldots, \phi_k$ corresponding to functions of the structure (3.2). In the implementation we represent these iterates in a polynomial basis using p vectors of length n and $k - p$ vectors of length r.

For the operator $\mathcal{F}\mathcal{B}$ there is in general no obvious scalar product to be used in the construction. We will, similar to [14], couple the scalar product with the polynomial representation, in the sense that we will use the Euclidean inner product corresponding to the coefficient vectors in the basis. Consider any polynomial basis g_0, g_1, ... and any two functions φ and ψ that can be expressed as

$$\varphi(\theta) = \sum_{i=0}^{\infty} g_i(\theta)x_i, \qquad \psi(\theta) = \sum_{i=0}^{\infty} g_i(\theta)y_i.$$

Then, we define the scalar product as follows

$$\langle \varphi, \psi \rangle := \sum_{i=0}^{\infty} y_i^* x_i.$$

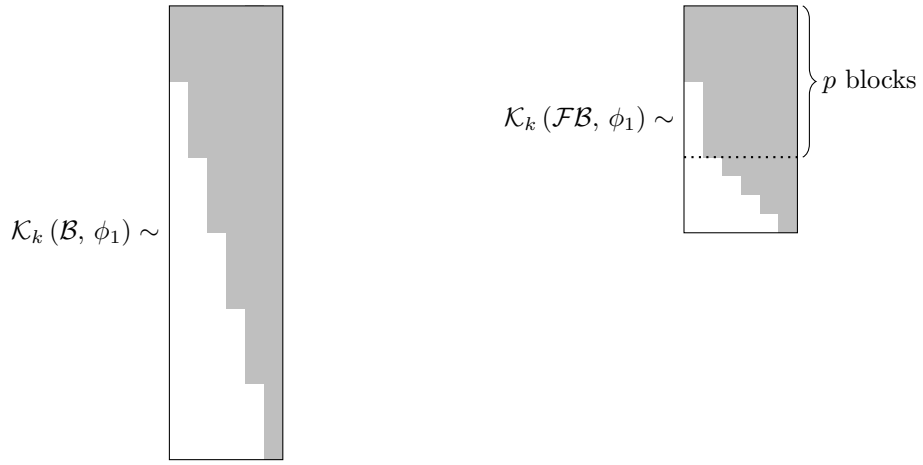
We will work out the algorithm for both the monomial basis as well as a (scaled) Chebyshev basis, although other choices of polynomial bases are also possible. The scalar product corresponding to the Chebyshev basis is known (from [14, Section 5.2-5.3]) to converge quickly for the spectral discretization of a differential operator for certain nonlinear eigenvalue problems.

By using the coupling of basis and scalar product, we can carry out the scalar product of two functions directly in the compressed representation, as can be seen from the following lemma.

Lemma 3.2. *Consider a polynomial basis $g_0, g_1, \ldots$ and suppose the vector-valued polynomials φ and ψ are given by $\varphi(\theta) = x_0 g_0(\theta) + \cdots + x_k g_k(\theta)$ and $\psi(\theta) = y_0 g_0(\theta) + \cdots + y_m g_m(\theta)$. Moreover, suppose the functions φ and ψ have the structure (3.2), i.e., $x_i = Q\hat{x}_i$, $y_i = Q\hat{y}_i$ when $i \geq p$. Then,*

$$\langle \varphi, \psi \rangle := \sum_{i=0}^{p-1} y_i^* x_i + \sum_{i=p}^{\min(k,m)} \hat{y}_i^* \hat{x}_i.$$

Figure 1 illustrates graphically the non-zero structure of the basis representations constructed by the standard infinite Arnoldi method [14] as well as by its low-rank version. Figure 1(a) shows that when we apply Arnoldi's method to $\mathcal{B}$ with constant starting function, the non-zero part of the basis grows by a block row consisting of n rows. In contrast to this, Figure 1(b) shows that when we apply Arnoldi's method to $\mathcal{F}\mathcal{B}$ with constant starting function, the basis matrix is only expanded by a block row consisting of r rows, since the vector coefficients for polynomials of degree higher than p can be represented with vectors of length r.



(a) Standard Infinite Arnoldi method (b) Low-rank Infinite Arnoldi method

Figure 1: Structure of the basis representations of the standard infinite Arnoldi method [14] and its low-rank version presented in this paper.

3.1 Taylor coefficient map

We wish to carry out Arnoldi's method for $\mathcal{FB}$ where the functions are represented in a polynomial basis. As a first step in this construction we need the action of $\mathcal{FB}$ in the monomial basis. The following theorem specifies $\mathcal{FB}$ for functions of the structure (3.2).

Theorem 3.3 (General coefficient map for $\mathcal{FB}$ in the monomial basis)**.** *Suppose that φ is given by*

$$\varphi(\theta) := \sum_{i=0}^{p-1} \theta^i x_i + \sum_{i=p}^{N-1} \theta^i Q\hat{x}_i,$$

where $x_0, \ldots, x_{p-1}, Q\hat{x}_p, \ldots, Q\hat{x}_{N-1} \in \mathbb{C}^n$ denote the vector coefficients in the monomial basis. Then, the coefficients of $\psi := \mathcal{FB}\varphi$, i.e.,

$$\psi(\theta) = (\mathcal{FB}\varphi)(\theta) = \sum_{i=0}^{p-1} \theta^i y_i + \sum_{i=p}^{N} \theta^i Q\hat{y}_i,$$

are given by

$$y_0 = \sum_{i=0}^{p-1} B_i x_i + \sum_{i=p}^{N-1} U_i \hat{x}_i, \tag{3.3}$$

and

$$\begin{aligned} \begin{bmatrix} y_1 & \cdots & y_{p-1} \end{bmatrix} &= \begin{bmatrix} x_0/1 & \cdots & x_{p-2}/(p-1) \end{bmatrix}, \\ \hat{y}_p &= Q^* x_{p-1}/p, \\ \begin{bmatrix} \hat{y}_{p+1} & \cdots & \hat{y}_N \end{bmatrix} &= \begin{bmatrix} \hat{x}_p/(p+1) & \cdots & \hat{x}_{N-1}/N \end{bmatrix}, \end{aligned} \tag{3.4}$$

Proof. This follows directly from the definitions of $\mathcal{F}$ and $\mathcal{B}$. □

In order to carry out the action in practice, some analysis is required for the specific problem in order to find an explicit and efficient expression for (3.3). Fortunately, we

can simplify somewhat, if the problem is expressed in terms of the coefficient matrices M_i of the original nonlinear eigenvalue problem (1.1), since simple manipulations yield

$$y_0 = -M_0^{-1}\left(\sum_{i=1}^{p-1} M_i y_i + \frac{1}{p} M_p x_{p-1} + \sum_{i=p+1}^{N} V_i \hat{y}_i\right), \tag{3.5}$$
$$= -M_0^{-1}\left(\sum_{i=0}^{p-1} \frac{1}{i+1} M_{i+1} x_i + \sum_{i=p}^{N-1} \frac{1}{i+1} V_{i+1} \hat{x}_i\right).$$

3.2 Chebyshev coefficient map

It was illustrated and explained in [14] that for certain problems it is natural and much more efficient to work with the inner product corresponding to the Chebyshev basis, in particular in terms of asymptotic convergence rate. We will now derive the coefficient map for $\mathcal{FB}$ in Chebyshev basis. In Section 5 we illustrate that we have a considerable improvement in performance for certain problems.

Let $T_0, T_1, \ldots$ be the Chebyshev polynomials of the first kind scaled to an interval $[a, b]$, i.e.,

$$T_i(\theta) = \cos(i \arccos(k\theta + c)) \tag{3.6}$$

where $c = \frac{a+b}{a-b}$ and $k = \frac{2}{b-a}$. We will need an explicit representation of the integration of a polynomial expressed in the Chebyshev basis. Let L_N correspond to this map, i.e., for any $N \in \mathbb{N}$ we have

$$\begin{bmatrix} T_0(\theta) \\ \vdots \\ T_{N-1}(\theta) \end{bmatrix} = L_N \begin{bmatrix} T_1'(\theta) \\ \vdots \\ T_N'(\theta) \end{bmatrix}.$$

Then, the matrix L_N is triangular and an explicit expression is given in [14, Equation 21]. We will partition L_N into blocks as follows

$$L_N =: \begin{bmatrix} L_{11} & 0 & 0 \\ L_{21} & L_{22} & 0 \\ L_{31} & L_{32} & L_{33} \end{bmatrix}, \tag{3.7}$$

where $L_{11} \in \mathbb{R}^{(p-1)\times(p-1)}$, $L_{22} \in \mathbb{R}$, and $L_{33} \in \mathbb{R}^{(N-p)\times(N-p)}$.

In the formulations of the coefficient map we will need the coefficients transforming a Chebyshev polynomial into its monomial coefficients. Let this matrix be given by $U \in \mathbb{R}^{N\times N}$, i.e.,

$$\begin{bmatrix} T_0(\theta) \\ T_1(\theta) \\ T_2(\theta) \\ \vdots \end{bmatrix} = \begin{bmatrix} u_{0,0} & 0 & 0 & 0 & \cdots \\ u_{1,0} & u_{1,1} & 0 & 0 & \cdots \\ u_{2,0} & u_{2,1} & u_{2,2} & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots & \ddots \end{bmatrix} \begin{bmatrix} 1 \\ \theta \\ \theta^2 \\ \vdots \end{bmatrix} =: U \begin{bmatrix} 1 \\ \theta \\ \theta^2 \\ \vdots \end{bmatrix}. \tag{3.8}$$

Moreover, let $v_p \in \mathbb{R}^p$ be defined as follows

$$v_p^T = \begin{bmatrix} u_{p,0} & u_{p,1} & \cdots & u_{p,p-1} \end{bmatrix} \begin{bmatrix} u_{0,0} & 0 & \cdots & 0 \\ u_{1,0} & u_{1,1} & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ u_{p-1,0} & u_{p-1,1} & \cdots & u_{p-1,p-1} \end{bmatrix}^{-1}. \tag{3.9}$$

In contrast to the monomial case, the application of $\mathcal{F}$ modifies all first $p+1$ coefficients when it is represented in the Chebyshev basis. More precisely, the coefficients are modified as follows.

Lemma 3.4 (The representation of the operator $\mathcal{F}$ in the Chebyshev basis)**.** *Suppose φ is given by*

$$\varphi(\theta) = \sum_{i=0}^{p} T_i(\theta) x_i + \sum_{i=p+1}^{N} T_i(\theta) Q \hat{x}_i, \tag{3.10}$$

with $x_0, \ldots, x_p \in \mathbb{C}^n$ and $\hat{x}_{p+1}, \ldots, \hat{x}_N \in \mathbb{C}^r$ and $T_0, T_1, \ldots$ are defined by (3.6)*. Then*

$$(\mathcal{F}\varphi)(\theta) = \sum_{i=0}^{p-1} T_i(\theta) y_i + \sum_{i=p}^{N} T_i(\theta) Q \hat{y}_i,$$

where

$$\begin{aligned} \begin{bmatrix} y_0 & \cdots & y_{p-1} \end{bmatrix} &= \begin{bmatrix} x_0 & \cdots & x_{p-1} \end{bmatrix} + (I - QQ^*) x_p v_p^T, \\ \hat{y}_p &= Q^* x_p, \\ \begin{bmatrix} \hat{y}_{p+1} & \cdots & \hat{y}_N \end{bmatrix} &= \begin{bmatrix} \hat{x}_{p+1} & \cdots & \hat{x}_N \end{bmatrix}, \end{aligned}$$

with v_p is defined by (3.9)*.*

Proof. Since the operator $\mathcal{F}$ is defined in the monomial basis, we transform φ (3.10) to the monomial basis, carry out the operation $\mathcal{F}$, and then transform it back to the Chebyshev basis. Let $X_1 = \begin{bmatrix} x_0 & \cdots & x_{p-1} \end{bmatrix}$ and $X_3 = \begin{bmatrix} \hat{x}_{p+1} & \cdots & \hat{x}_N \end{bmatrix}$. We have

$$\begin{aligned} \varphi(\theta) &= \begin{bmatrix} X_1 & x_p & QX_3 \end{bmatrix} \begin{bmatrix} T_0(\theta) & \cdots & T_N(\theta) \end{bmatrix}^*, \\ &= \begin{bmatrix} X_1 & x_p & QX_3 \end{bmatrix} U \begin{bmatrix} 1 & \theta & \cdots & \theta^N \end{bmatrix}^*, \end{aligned} \tag{3.11}$$

where U is defined in (3.8). We now partition U according to

$$U := \begin{bmatrix} U_{11} & 0 & 0 \\ U_{21} & U_{22} & 0 \\ U_{31} & U_{32} & U_{33} \end{bmatrix},$$

where $U_{11} \in \mathbb{R}^{p \times p}$, $U_{22} \in \mathbb{R}$, and $U_{33} \in \mathbb{R}^{(N-p)\times(N-p)}$. By carrying out the multiplication in (3.11) we have

$$\varphi(\theta) = \begin{bmatrix} X_1 U_{11} + x_p U_{21} + QX_3 U_{31} & x_p U_{22} + QX_3 U_{32} & QX_3 U_{33} \end{bmatrix} \begin{bmatrix} 1 & \theta & \cdots & \theta^N \end{bmatrix}^*.$$

Recall that $\mathcal{F}$ corresponds to multiplying all monomial coefficients of degree equal or higher than p by QQ^*. By using that Q has orthonormal columns, we have that

$$\begin{aligned} (\mathcal{F}\varphi)(\theta) &= \begin{bmatrix} X_1 U_{11} + x_p U_{21} + QX_3 U_{31} & QQ^* x_p U_{22} + QX_3 U_{32} & QX_3 U_{33} \end{bmatrix} \begin{bmatrix} 1 \\ \vdots \\ \theta^N \end{bmatrix}, \\ &= \begin{bmatrix} X_1 + (I - QQ^*) x_p v_p^T & QQ^* x_p & QX_3 \end{bmatrix} \begin{bmatrix} U_{11} & 0 & 0 \\ U_{21} & U_{22} & 0 \\ U_{31} & U_{32} & U_{33} \end{bmatrix} \begin{bmatrix} 1 \\ \vdots \\ \theta^N \end{bmatrix}, \end{aligned}$$

where $v_p^T = U_{21} U_{11}^{-1}$. Finally, noting that

$$U \begin{bmatrix} 1 & \theta & \cdots & \theta^N \end{bmatrix}^* = \begin{bmatrix} T_0(\theta) & \cdots & T_N(\theta) \end{bmatrix}^*.$$

completes the proof. □

By combining the lemma characterizing the coefficient map corresponding to $\mathcal{F}$ we can now derive the coefficient map corresponding to $\mathcal{FB}$.

Theorem 3.5 (General coefficient map for $\mathcal{FB}$ in the Chebyshev basis)**.** *Suppose φ is given by*

$$\varphi(\theta) := \sum_{i=0}^{p-1} T_i(\theta) x_i + \sum_{i=p}^{N-1} T_i(\theta) Q \hat{x}_i,$$

where $T_0, T_1, \dots$ are defined by (3.6) *and the columns of $X \in \mathbb{C}^{n \times N}$*

$$X := \begin{bmatrix} X_1 & x_{p-1} & QX_3 \end{bmatrix},$$

denote the vector coefficients of φ, i.e., $X_1 := \begin{bmatrix} x_0 & \cdots & x_{p-2} \end{bmatrix}$ and $X_3 := \begin{bmatrix} \hat{x}_p & \cdots & \hat{x}_{N-1} \end{bmatrix}$. Then, the expansion of $\psi := \mathcal{FB}\varphi$, i.e.,

$$\psi(\theta) = (\mathcal{FB}\varphi)(\theta) = \sum_{i=0}^{p-1} T_i(\theta) y_i + \sum_{i=p}^{N} T_i(\theta) Q \hat{y}_i,$$

is given by

$$y_0 = \left(\sum_{i=0}^{p-1} \left(B\left(\frac{d}{d\theta}\right) T_i(\theta) x_i \right) + \sum_{i=p}^{N-1} \left(B\left(\frac{d}{d\theta}\right) T_i(\theta) Q \hat{x}_i \right) \right)_{\theta=0} - \begin{bmatrix} X_1 & x_{p-1} & QX_3 \end{bmatrix} L_N \begin{bmatrix} T_1(0) \\ \vdots \\ T_N(0) \end{bmatrix} + (I - QQ^*) x_{p-1} v_{p,1}, \tag{3.12}$$

and

$$\begin{aligned} \begin{bmatrix} y_1 & \cdots & y_{p-1} \end{bmatrix} &= X_1 L_{11} + x_{p-1} L_{21} + QX_3 L_{31} + (I - QQ^*) x_{p-1} L_{22} \tilde{v}_p^T, \\ y_p &= Q^* x_{p-1} L_{22} + X_3 L_{32}, \\ \begin{bmatrix} \hat{y}_{p+1} & \cdots & \hat{y}_N \end{bmatrix} &= X_3 L_{33}, \end{aligned} \tag{3.13}$$

where L_{ij} are defined by (3.7) *and we denote $v_p^T =: (v_{p,1}, \tilde{v}_p^T)$ with $v_{p,1} \in \mathbb{R}$, $\tilde{v}_p \in \mathbb{R}^{p-1}$.*

Proof. The proof consists of two parts. Firstly, we will use the general coefficient map defined in [14, Theorem 4] in order to obtain the coefficients of $\mathcal{B}\varphi$. Next, we apply Lemma 3.4 resulting in the coefficients of $\psi := \mathcal{FB}\varphi$.

Let $Z := \begin{bmatrix} z_0 & Z_1 & z_p & QZ_3 \end{bmatrix}$ with $Z_1 := \begin{bmatrix} z_1 & \cdots & z_{p-1} \end{bmatrix}$ and $Z_3 := \begin{bmatrix} z_{p+1} & \cdots & z_N \end{bmatrix}$ denote the coefficients of the function $\mathcal{B}\varphi$, i.e.,

$$(\mathcal{B}\varphi)(\theta) = \sum_{i=0}^{p} T_i(\theta) z_i + \sum_{i=p+1}^{N} T_i(\theta) Q z_i.$$

Then, from the general coefficient map defined in [14, Theorem 4], we have

$$\begin{bmatrix} Z_1 & z_p & QZ_3 \end{bmatrix} = \begin{bmatrix} X_1 & x_{p-1} & QX_3 \end{bmatrix} \begin{bmatrix} L_{11} & 0 & 0 \\ L_{21} & L_{22} & 0 \\ L_{31} & L_{32} & L_{33} \end{bmatrix},$$

which yields

$$Z_1 = X_1 L_{11} + x_{p-1} L_{21} + QX_3 L_{31}, \tag{3.14a}$$

$$z_p = x_{p-1} L_{22} + QX_3 L_{32}, \tag{3.14b}$$

$$Z_3 = X_3 L_{33}. \tag{3.14c}$$

An explicit expression for the vector $z_0 \in \mathbb{C}^n$ can be found by noting that [14, Equation 22] can be rephrased using [14, Equation 12] and specialized for φ. This leads to

$$z_0 = \sum_{i=0}^{p-1} \left(B(\frac{d}{d\theta}) T_i(\theta) x_i \right)(0) + \sum_{i=p}^{N-1} \left(B(\frac{d}{d\theta}) T_i(\theta) Q\hat{x}_i \right)(0) - \begin{bmatrix} X_1 & x_{p-1} & QX_3 \end{bmatrix} L_N \begin{bmatrix} T_1(0) \\ \vdots \\ T_N(0) \end{bmatrix}. \tag{3.15}$$

To complete the proof, we apply Lemma 3.4 which results in $y_0 = z_0 + (I - QQ^*)x_{p-1}v_{p,1}$ and (3.12)–(3.13). □

Similar to the monomial case we need to find an efficient, accurate and explicit expressions for y_0 in (3.12). The difficulty in deriving such an expression should not be underestimated. Unfortunately, unlike the monomial case, in particular (3.5), expressing y_0 in terms of the coefficients of the original nonlinear eigenvalue problem $M_0, M_1, \ldots$ does not considerably simplify the problem, although a general approach based on manipulations similar to [14, Appendix A] is feasable but somewhat tedious in general.

In this work we will derive explicit expressions for an important situation. We consider delay eigenvalue problems and specialize the result as follows. Suppose

$$M(\lambda) = -\lambda I + A_0 + A_1 e^{-\tau\lambda}. \tag{3.16}$$

where $A_1 = VQ^*$, such that we can set $p = 1$. Such problems occurs in the stability analysis of PDE with pointwise delayed feedback, as we shall further illustrate in Section 5.2. The coefficient map (Theorem 3.5) simplifies as follows for (3.16).

Corollary 3.6 (Delay eigenvalue problem with single delay and $p = 1$)**.** *Consider the nonlinear eigenvalue problem* (1.1) *where* $M(\lambda)$ *is given by* (3.16) *with* $A_1 = VQ^*$ *and* $Q \in \mathbb{C}^{n\times r}$ *has orthonormal columns. Let* T_0*,* T_1*, ... be the Chebyshev polynomials of the first kind* (3.6) *scaled to the interval* $(a, b) = (-\tau, 0)$*. Moreover, suppose* φ *is given by*

$$\varphi(\theta) = T_0(\theta)x_0 + \sum_{i=1}^{N-1} T_i(\theta) Q\hat{x}_i, \tag{3.17}$$

where $x_0 \in \mathbb{C}^n$ *and* $x_1, x_2, \ldots \in \mathbb{C}^r$*. Then, the expansion of* $\psi := \mathcal{F}\mathcal{B}\varphi$*, i.e.,*

$$\psi(\theta) = (\mathcal{F}\mathcal{B}\varphi)(\theta) = T_0(\theta)y_0 + \sum_{i=1}^{N} T_i(\theta) Q y_i,$$

is given by

$$y_0 = (A_0 + A_1)^{-1} \left(x_0 + Q \sum_{i=1}^{N-1} \hat{x}_i - A_0 \left[z_1 + Q \sum_{i=2}^{N} \hat{y}_i \right] - A_1 \left[z_1 T_1(-\tau) + Q \sum_{i=2}^{N} T_i(-\tau)\hat{y}_i \right] \right) + \frac{\tau}{2}(I - QQ^*)x_0,$$

and

$$\begin{bmatrix} y_1 & \cdots & y_N \end{bmatrix} = \begin{bmatrix} Q^*x_0 & x_1 & \cdots & x_{N-1} \end{bmatrix} L_N, \tag{3.18}$$

with

$$z_1 = \begin{cases} \frac{\tau}{2}x_0 - \frac{\tau}{4}Qx_2 & N \geq 3 \\ \frac{\tau}{2}x_0 & \textit{otherwise} \end{cases}. \tag{3.19}$$

Proof. Note that when $p = 1$, several matrices in Theorem 3.5 should be interpreted as empty matrices, in particular L_{11}, L_{21}, L_{31}, Z_1 and X_1, implying that $\begin{bmatrix} y_1 & \cdots & y_N \end{bmatrix}$ can be directly computed from (3.18).

The formula for y_0 is simplified if we introduce the variable z_1 in (3.14b). Due to the partitioning of L_N in (3.7) and the explicit formula for L_N in [14, Equation 21] with $(a, b) = (-\tau, 0)$ we have in our case that $L_{22} = \tau/2$ and $L_{32} = \begin{pmatrix} 0 & -\frac{\tau}{4} & 0 & \cdots & 0 \end{pmatrix}^T$. Hence, the definition of z_1 in (3.14b) simplifies to (3.19).

We derive the formula of y_0 from the fact that $y_0 = z_0 + (I - QQ^*)z_1$, similar to the last step in the proof of Theorem 3.5. In our setting $v_p = 1$ and $z_1 = x_0 L_{22}$ with $L_{22} = 2(b - a)/4 = \tau/2$ from (3.7). The expression for z_0 is found by inserting the definition of $M(\lambda)$ and $B(\lambda)$ into (3.15). □

3.3 Low-rank infinite Arnoldi method

From Section 3.1 and Section 3.2 we now know how to compute the action of $\mathcal{FB}$ in monomial basis or Chebyshev basis for functions with the structure (3.2). Lemma 3.2 suggests a natural way to carry out the scalar product for such functions. The action of the operator and the scalar product are the only ingredients needed in order to carry out Arnoldi's method in an operator setting, i.e., [14, Algorithm 1].

As usual for the Arnoldi method, we will denote the upper block of the rectangular Hessenberg matrix $\underline{H}_k \in \mathbb{C}^{(k+1)\times k}$ by $H_k \in \mathbb{C}^{k\times k}$ and the (i, j) element of $\underline{H}_k$ is denoted $h_{i,j}$. We summarize the algorithm in Algorithm 1 and Algorithm 2, where we have separated the algorithm into two parts in order to simplify the presentation of the restarting in the following sections. We will for reasons of efficiency stack the coeffients into vectors and matrices such that the orthogonalization can be carried out with simple operations on larger matrices and vectors as illustrated in Figure 1.

Algorithm 1: Low-rank infinite Arnoldi method

Input : $V_1 = v \in \mathbb{C}^n$, k is number of steps
Output: Reciprocal Ritz values $\tilde{\lambda}_1, \ldots, \tilde{\lambda}_k$

1 Set $\underline{H}_0 = 0$,
for $j = 1, 2, \ldots, k$ **do**
2 | $[V_{j+1}, \underline{H}_j]$ = Algorithm 2 with input $[V_j, \underline{H}_{j-1}]$
end
3 Return approximate solutions of (2.1) $\tilde{\lambda}_j = 1/\mu_j$ where $\mu_j \in \sigma(H_k)$, $j = 1, \ldots, k$.

Remark 3.7 (Extraction of eigenvectors)**.** *The result of Algorithm 1 is the matrix H_k and V_k. The approximate eigenvalues $\tilde{\lambda}_j$ are the reciprocal eigenvalues of H_k. We also need to form approximations of the corresponding eigenvectors v_j. With V_k we have a representation of an approximate eigenfunctions of $\mathcal{FB}$. We propose here to compute approximate eigenvectors by function φ at $\theta = 0$, since exact eigenfunctions satisfies $\varphi(0) = v_j$ (according to Theorem 2.1). In the Taylor version (Section 3.1), this corresponds to using the first n rows of V_k, whereas for the Chebyshev version (Section 3.2) we make the corresponding evaluation by computing $T_i(0)$.*

Algorithm 2: Low-rank infinite Arnoldi step

Input : V_j, $\underline{H}_{j-1}$
Output: V_{j+1}, $\underline{H}_j$

1 Let $\begin{bmatrix} x_0^* & \cdots & x_{p-1}^* & \hat{x}_p^* & \cdots & \hat{x}_j^* \end{bmatrix}^* := v_j^*$.
2 Compute $y_0, \ldots, y_{p-1} \in \mathbb{C}^n$ and $\hat{y}_p, \hat{y}_{p+1}, \ldots, \hat{y}_{j+1} \in \mathbb{C}^r$ by either

(a) the Taylor coefficient map according to (3.4)–(3.5); or
(b) the Chebyshev coefficient map according to (3.12)–(3.13).

3 Let $w_j := \begin{bmatrix} y_0^* & y_1^* & \cdots & y_{p-1}^* & \hat{y}_p^* & \hat{y}_{p+1}^* & \cdots & \hat{y}_{j+1}^* \end{bmatrix}^*$.
4 Expand V_j with one block row with n or r rows.
5 Orthogonalize $\hat{w}_j := w_j - V_j h_j$, where $h_j = V_j^* w_j$.
6 Compute $\beta_j = \|\hat{w}_j\|_2$ and let $v_{j+1} = \hat{w}_j / \beta_j$.
7 Let $\underline{H}_j = \begin{bmatrix} \underline{H}_{j-1} & h_j \\ 0 & \beta_j \end{bmatrix} \in \mathbb{C}^{(j+1)\times j}$.
8 Expand V_j into $V_{j+1} = \begin{bmatrix} V_j & v_{j+1} \end{bmatrix}$.

4 Implicit restarting and locking

In the analysis, implementation and enhancement of the Arnoldi method, we often use that the result of the Arnoldi method satisfies a so-called Arnoldi relation [9, Equation 10.5.2]. Due to the fact that Algorithm 1 is equivalent to the Arnoldi method applied to the operator $\mathcal{FB}$, the result will also satisfy an Arnoldi relation. More precisely, the function-setting Arnoldi relation generated by Algorithm 1 can be formulated as follows. Let $\Phi_j(\theta) \in \mathbb{C}^{n\times j}$ correspond to the matrix consisting of columns $\varphi_1(\theta), \ldots, \varphi_j(\theta)$, i.e.,

$$\Phi_j(\theta) := \begin{bmatrix} \varphi_1(\theta) & \cdots & \varphi_j(\theta) \end{bmatrix}.$$

Then, the output of Algorithm 1 satisfies

$$(\mathcal{FB}\,\Phi_j)(\theta) = \Phi_{j+1}(\theta)\,\underline{H}_j, \tag{4.1}$$

where we can express Φ_j explicitly as

$$\Phi_j(\theta) = \begin{bmatrix} (q_0(\theta), \ldots, q_{p-1}(\theta)) \otimes I_n & (q_p(\theta), \ldots, q_{j-1}(\theta)) \otimes Q \end{bmatrix} V_j, \tag{4.2}$$

and $q_i(\theta) = \theta^i$ or $q_i(\theta) = \hat{T}_i(\theta)$, $i = 0, \ldots, j$ depending on which basis is used.

We will now see that as a consequence of the fact that we have an Arnoldi relation (4.1), we will be able to carry out implicit restarting very similar to the implicit restarting procedures for the standard Arnoldi method. The restarting can be seen as a procedure to (essentially) compress the Arnoldi relation resulting in a basis matrix with a smaller number of columns. Due to the fact that the infinite Arnoldi method (Algorithm 1) has a growth also in the height of the basis matrix, we will have a growth with each restart. However, if r is small, this growth is moderate and the growth in the height of the basis matrix is not restrictive. This is illustrated in Figure 2.

4.1 Krylov-Schur style implicit restarting and locking

In order to carry out implicit restarting and locking easily, we will work with a Krylov–Schur recurrence relation [22] in every iteration of the algorithm. This can be achieved by computing a Schur decomposition of H_j

$$H_j = Z_j S_j Z_j^*, \tag{4.3}$$

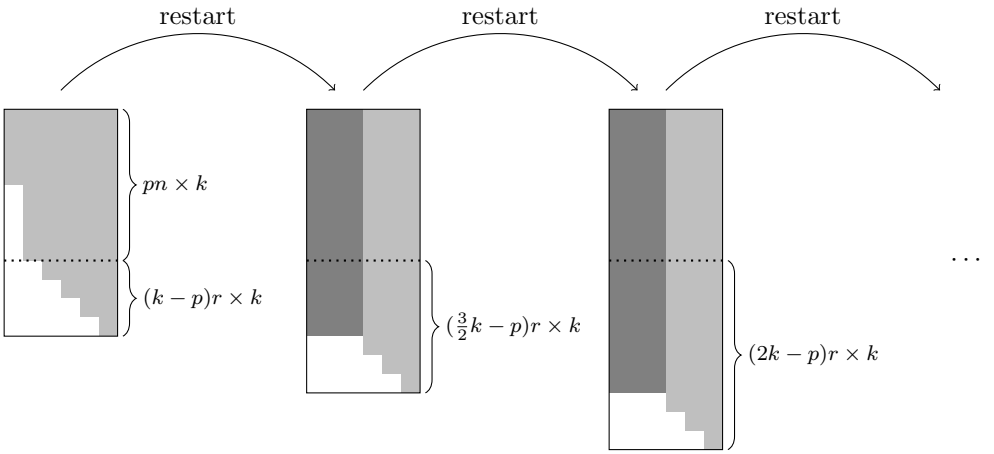


Figure 2: Graphical illustration of implicit restarting on the Krylov subspace $\mathcal{K}_k(\mathcal{FB}, \varphi_1)$ for the case $p = 2$. The light grey shade areas represent the nonzero structure of the growing coefficient vectors, whereas the dark grey shade represent the nonzero structure of the coefficient vectors after an implicit restart.

where S_j is an upper quasitriangular matrix and $Z_j^* Z_j = I_j$. Using the Schur decomposition (4.3), we can transform (4.1) into the following Krylov–Schur recurrence relation

$$(\mathcal{FB}\,\Psi_j)(\theta) = \Psi_{j+1}(\theta)\,\underline{S}_j, \tag{4.4}$$

where

$$\Psi_{j+1}(\theta) := \begin{bmatrix}\Phi_j(\theta)Z_j & \varphi_{j+1}\end{bmatrix}, \tag{4.5}$$

and

$$\underline{S}_j := \begin{bmatrix} S_j \\ h_{j+1}^* Z_j \end{bmatrix}.$$

Let S_j be an ordered Schur decomposition of H_k

$$S_j = Z_j^* H_j Z_j = \begin{bmatrix} Z_1 & Z_2 & Z_3 \end{bmatrix}^* H_j \begin{bmatrix} Z_1 & Z_2 & Z_3 \end{bmatrix} = \begin{bmatrix} S_{11} & S_{12} & S_{13} \\ & S_{22} & S_{23} \\ & & S_{33} \end{bmatrix}, \tag{4.6}$$

where $S_{11} \in \mathbb{C}^{\ell\times\ell}$, $S_{22} \in \mathbb{C}^{(m-\ell)\times(m-\ell)}$, and $S_{33} \in \mathbb{C}^{(k-m)\times(k-m)}$ are upper quasitriangular matrices. The ordering is as follows: the eigenvalues of S_{11}, S_{22}, and S_{33} are, respectively, the very accurate Ritz values, the wanted but not yet converged Ritz values, and the unwanted Ritz values. Hence,

$$\underline{S}_j = \begin{bmatrix} S_{11} & S_{12} & S_{13} \\ & S_{22} & S_{23} \\ & & S_{33} \\ b_1^* & b_2^* & b_3^* \end{bmatrix}, \tag{4.7}$$

where $b_i^* = h_{j+1}^* Z_i$, for $i = 1, 2, 3$.

Note that by using the ordered Schur decomposition of H_j (4.6)

$$(\mathcal{FB}) \begin{bmatrix}\Psi_1(\theta) & \Psi_2(\theta)\end{bmatrix} = \begin{bmatrix}\Psi_1(\theta) & \Psi_2(\theta) & \psi_{j+1}(\theta)\end{bmatrix} \begin{bmatrix} S_{11} & S_{12} \\ 0 & S_{22} \\ b_1^* & b_2^* \end{bmatrix},$$

where

$$\Psi_1(\theta) := \begin{bmatrix}\psi_1(\theta) & \cdots & \psi_\ell(\theta)\end{bmatrix},$$
$$\Psi_2(\theta) := \begin{bmatrix}\psi_{\ell+1}(\theta) & \cdots & \psi_m(\theta)\end{bmatrix},$$

is also a Krylov–Schur decomposition. Thus, the purging problem can be solved by moving the unwanted Ritz values into the southeast corner of the matrix S and truncating the decomposition. Next, the Arnoldi process is restarted.

The use of the ordered Schur decomposition has also the advantage that deflation and locking of the converged Ritz pairs corresponds to setting $b_1^* = 0$ in (4.7), yielding the following recurrence relation

$$(\mathcal{FB})\begin{bmatrix}\Psi_1(\theta) & \Psi_2(\theta)\end{bmatrix} = \begin{bmatrix}\Psi_1(\theta) & \Psi_2(\theta) & \psi_{j+1}(\theta)\end{bmatrix}\begin{bmatrix}S_{11} & S_{12}\\ 0 & S_{22}\\ 0 & b_2^*\end{bmatrix} + O(\|b_1\|).$$

Note that b_1 is a measure for the (unstructured) backward error of the corresponding eigenvalues of S_{11}. Hence, $\|b_1\|$ will be zero if the eigenvalues of S_{11} are exact. For more information, we refer to [17].

4.2 Implicit restarting and locking of Algorithm 1

The operations outlined in the previous section can now be combined with Algorithm 1 and Algorithm 2. In (4.5) we multiply $\Phi_j(\theta)$ by Z_j from the left. Note that, due to relation (4.2), this corresponds to multiplying the basis matrix V_j from the left by Z_j. We provide the algorithm details in Algorithm 3.

Algorithm 3: Implicit Restarted Infinite Arnoldi (IRIA) method

Input : $x_0 \in \mathbb{R}^n$, $k, m \in \mathbb{N}$
Output: Reciprocal Ritz values $\tilde{\lambda}_1, \ldots, \tilde{\lambda}_k$

Let $V_1 = x_0/\|x_0\|_2$ and $\underline{H}_0 = [\,]$.
for $j = 1, 2, \ldots$ **do**
  Compute V_{j+1} and $\underline{H}_j$ by Algorithm 2 based on V_j and $\underline{H}_{j-1}$.
  Compute the ordered Schur factorization of H_j according to (4.6).
  Let $\begin{bmatrix}b_1^* & b_2^* & b_3^*\end{bmatrix} := h_{j+1}^*\begin{bmatrix}Z_1 & Z_2 & Z_3\end{bmatrix}$.
  Let $\begin{bmatrix}U_1 & U_2 & U_3\end{bmatrix} := \begin{bmatrix}V_1 & V_2 & V_3\end{bmatrix} Z_j$.
  if $j > m$ **and** $\mathrm{mod}(j-k, m) = 0$ **then**
    Let $V_{j+1} = \begin{bmatrix}U_1 & U_2 & v_{j+1}\end{bmatrix}$.
    Let $\underline{H}_j = \begin{bmatrix}S_{11} & S_{12}\\ 0 & S_{22}\\ 0 & b_2^*\end{bmatrix}$.
  else
    Let $V_{j+1} = \begin{bmatrix}U_1 & U_2 & U_3 & v_{j+1}\end{bmatrix}$.
    Let $\underline{H}_j = \begin{bmatrix}S_{11} & S_{12} & S_{13}\\ 0 & S_{22} & S_{23}\\ 0 & 0 & S_{33}\\ 0 & b_2^* & b_3^*\end{bmatrix}$.
  end
end

5 Numerical experiments

Before presenting the results of the numerical experiments, we first introduce the following notation in order to simplify referencing to the different variants of the algorithms.

- T$\mathcal{B}$: Taylor variant for operator $\mathcal{B}$, i.e., Taylor variant of [14, Algorithm 2],
- T$\mathcal{FB}$: Taylor variant for operator $\mathcal{FB}$, i.e., Taylor variant of Algorithm 2,
- C$\mathcal{B}$: Chebyshev variant for operator $\mathcal{B}$, i.e., Chebyshev variant of [14, Algorithm 2],
- C$\mathcal{FB}$: Chebyshev variant for operator $\mathcal{FB}$, i.e., Chebyshev variant of Algorithm 2.

We will also add a R to denote the implicitly restarted variant, e.g., T$\mathcal{FB}$R and C$\mathcal{FB}$R denote respectively the implicitly restarted Taylor and Chebyshev variants with low-rank exploitation of Algorithm 3.

5.1 A random example

We illustrate the generality and efficiency of the algorithm by applying it to a problem with randomly generated matrices. Suppose

$$M(\lambda) = A_0 + \lambda A_1 + A_2\lambda^4 + A_3\sin(\lambda), \tag{5.1}$$

where $A_0, A_2 \in \mathbb{R}^{n\times n}$ are random sparse matrices with normal-distributed elements, $A_1 = -I$, and $A_3 = UQ^T$ with $U, Q \in \mathbb{R}^{n\times 2}$ randomly generated matrices and $Q^TQ = I$. For illustrative reasons we set $n = 1000$. In order to make the results reproducible, we have made the matrices available online*.

For this example, it is natural to select $p = 4$ and the expansions (1.2)–(1.3) are explicitly given by

$$\begin{aligned} M_0 &= A_0 & M_3 &= -UQ^T \\ M_1 &= A_1 + UQ^T & M_4 &= 4!\cdot A_2 \\ M_2 &= 0 & M_i &= V_iQ^T, \qquad i \geq 5 \end{aligned}$$

where

$$\begin{aligned} V_{2k+1} &= (-1)^k U, \qquad k \geq 2, \\ V_{2k} &= 0, \qquad\qquad k \geq 3. \end{aligned}$$

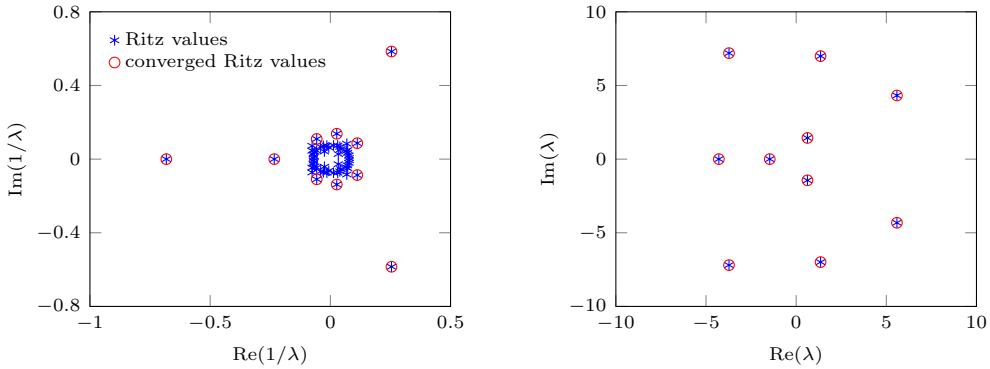
The goal in this experiment is to compute the 10 eigenvalues closest to the origin. For measuring the convergence of an approximate eigenpair (λ, x), we used the following relative residual norm

$$E(\lambda, x) = \frac{\|M(\lambda)x\|_2/\|x\|_2}{\|A_0\|_1 + |\lambda| + \|A_2\|_1|\lambda|^4 + \|A_3\|_1|\sin(\lambda)|}.$$

In the implementation, we precompute the LU-factorization of M_0 in order to use in the formula for y_0, given by (3.5), and the terms involving M_1 and M_3 are computed as follows: $M_1y_1 = A_1y_1 + V(Q^Ty_1)$ and $M_3y_3 = -V(Q^Ty_1)$, respectively.

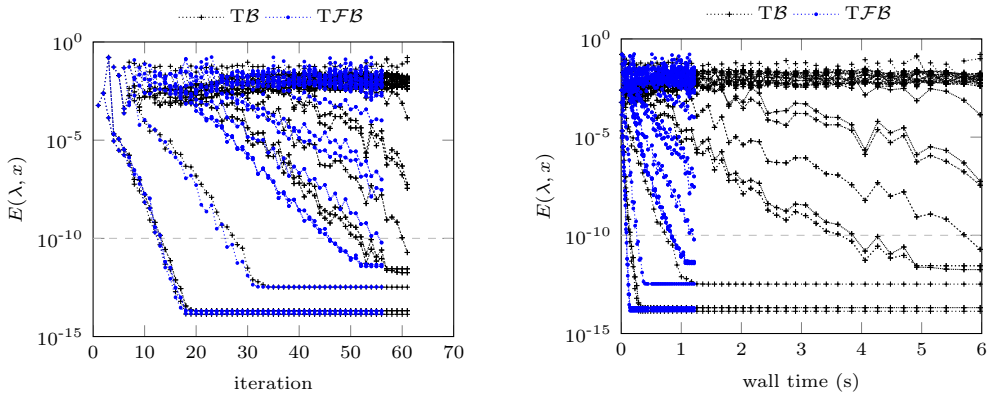
We first solved the nonlinear eigenvalue problem (5.1) by the variants T$\mathcal{B}$ and T$\mathcal{FB}$. The eigenvalues and results of this experiment are shown in Figures 3 and 4, respectively. We observe in Figure 4(a) that the application of the operator $\mathcal{F}$ has very little impact on the approximations generated by each iteration of the two variants. On the other hand, using a compressed representation for T$\mathcal{FB}$ as illustrated in Figure 1(b), gives a significant performance improvement in the sense that each iteration can be carried out

*`http://www.math.kth.se/~eliasj/src/lowranknep/example1_matrices_final.mat`



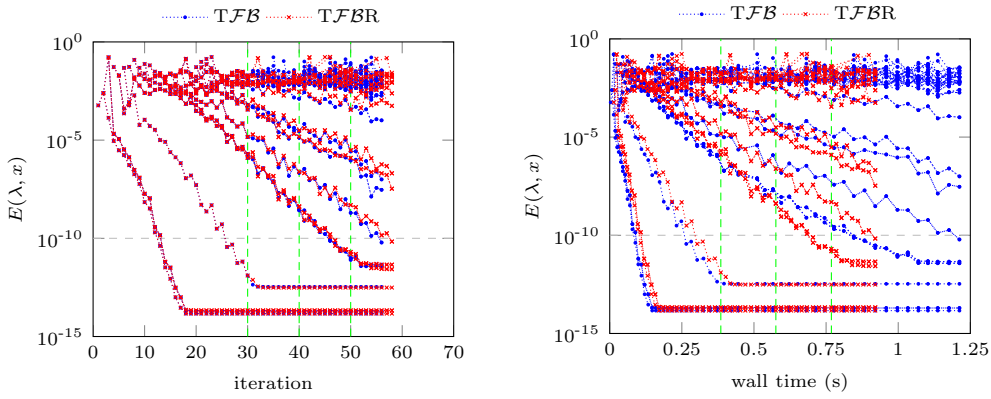
(a) Ritz values (b) Eigenvalues

Figure 3: Ritz values and approximate eigenvalues, i.e., reciprocal Ritz values, of the random example computed with variant T$\mathcal{FB}$. An eigenvalue is classified as converged if $E(\lambda, x) \leq 10^{-10}$.



(a) Iteration vs relative residual (b) Wall time vs relative residual

Figure 4: Comparison of T$\mathcal{B}$ and T$\mathcal{FB}$ for (5.1). The variants generate very similar results per iteration, but the computation time grows much faster for T$\mathcal{B}$ than for T$\mathcal{FB}$.



(a) Iteration vs relative residual (b) Wall time vs relative residual

Figure 5: Comparison of T$\mathcal{FB}$ and T$\mathcal{FB}$R for (5.1). The convergence is only slightly slowed down by the restart (in terms of result per iteration), whereas the computation is further reduced. The vertical green dashed lines indicate the restarts.

in less computation time. As shown in Figure 4(b), this results in a much lower total computation time for T$\mathcal{FB}$ compared to T$\mathcal{B}$.

Next, we solved (5.1) by the implicitly restarted variant T$\mathcal{FB}$R. The results are illustrated in Figure 5. We observe in Figure 5(a) that the convergence speed as a function of iteration is slightly worsened by the restarting. But as expected from restarting, we notice in Figure 5(b) that the convergence speed as a function of computation time is further improved.

We finally illustrate the growth of the computation time of the restarted variants as a function of iteration in Figure 6. As a comparison, the computation time grows slower when exploiting the structure. For this regime, we observe in Figure 6 that after the first restart the computation time grows essentially linearly with the iteration for T$\mathcal{FB}$R, and still superlinearly for T$\mathcal{B}$.

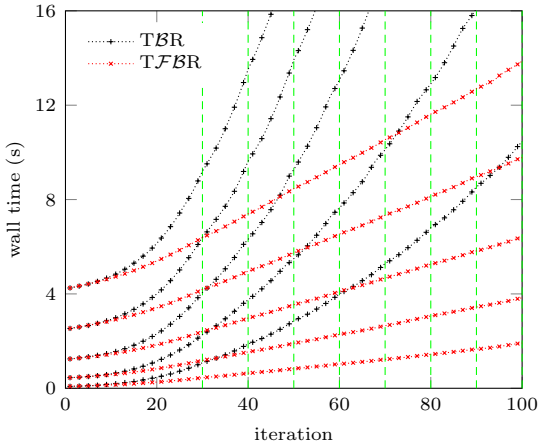


Figure 6: The computation time of T$\mathcal{B}$R and T$\mathcal{FB}$R for (5.1) as a function of iteration for $n = 1000, 2000, 3000, 4000, 5000$. The total computation time for T$\mathcal{FB}$R is considerably lower than for T$\mathcal{B}$R for larger n. The vertical green dashed lines indicate the restarts.

5.2 A delay eigenvalue problem

We model a one-dimensional clamped beam and delayed feedback control localized at the endpoint with a partial delay-differential equation. See [31, 10] for PDEs with delays. More precisely, we consider the one-dimensional DDE

$$u_t(x,t) = u_{xx}(x,t) + \delta(x-0.5)u(0.5,t-\tau),$$

with boundary conditions $u(0,t) = u_x(1,t) = 0$ and $\delta(x)$ a dirac impulse such that the problem corresponds to delayed pointwise feedback at $x = 0.5$. The finite difference discretization with n intervals results in the following delay eigenvalue problem

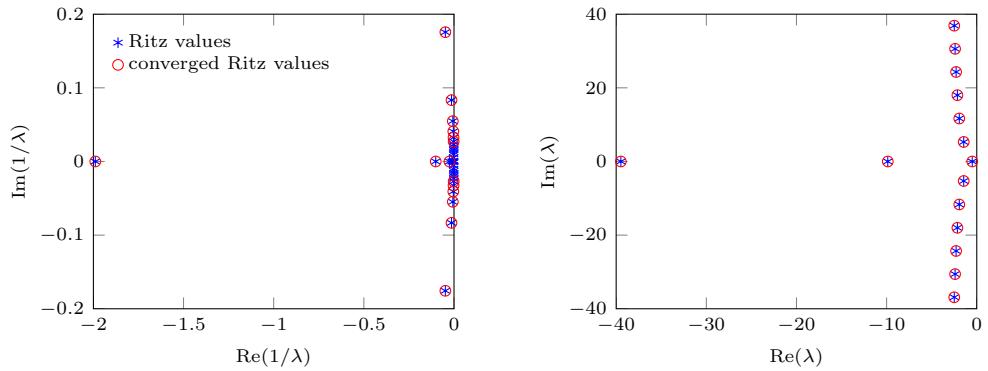
$$M(\lambda) = \lambda I + A_0 + A_1 e^{-\tau\lambda}, \tag{5.2}$$

where $A_0 \in \mathbb{C}^{n\times n}$ is a tridiagonal matrix and A_1 a rank 1 matrix. The goal in this experiment is to compute the 15 eigenvalues closest to the origin. For measuring the convergence of an approximate eigenpair (λ, x), we used the following relative residual norm

$$E(\lambda,x) = \frac{\|M(\lambda)x\|_2/\|x\|_2}{|\lambda| + \|A_0\|_1 + \|A_1\|_1|e^{-\tau\lambda}|}.$$

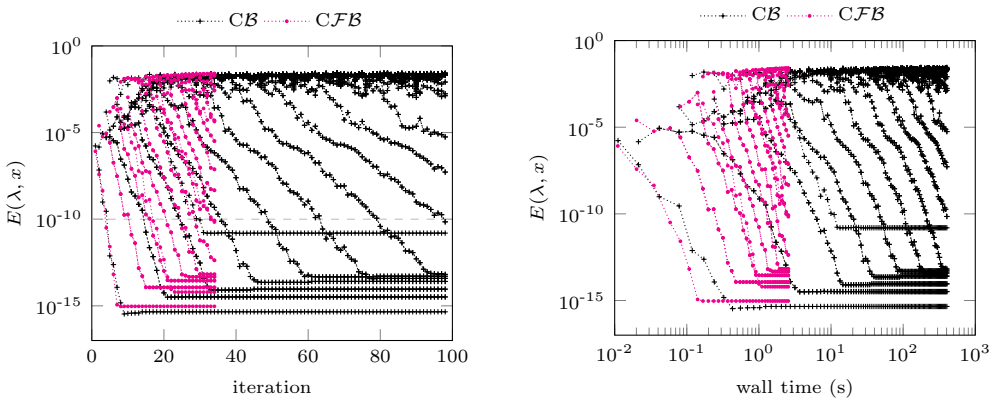
In every iteration of Algorithm 2, y_0 is computed with the formulas of Corollary 3.6.

The delay eigenvalue problem (5.2) with $n = 10.001$ is solved by variants C$\mathcal{B}$ and C$\mathcal{FB}$. Figure 7 shows the eigenvalues and the advantage of using the operator $\mathcal{F}$ together with the compact representation is reported Figure 8. If we only consider the relative error as a function of the iteration, we observe in Figure 8(a) that the eigenvalues converge faster for C$\mathcal{FB}$ than for C$\mathcal{B}$. But the major advantage of the compact representation can be seen in Figure 8(b), where we compare the relative error generated by C$\mathcal{B}$ and C$\mathcal{FB}$ as a function of wall time. In this figure we see that the total computation time for C$\mathcal{FB}$ is several orders of magnitude less than for C$\mathcal{B}$. As illustrated in Figure 1(a), the subspace vectors grow in every iteration of variant C$\mathcal{B}$ with a block of size $n = 10.001$. On the other hand, in variant C$\mathcal{FB}$ they grow after the first iteration only with blocks of size $r = 1$. Therefore, C$\mathcal{B}$ handles in this experiment with vectors of size $O(10^6)$, whether C$\mathcal{FB}$ only deals with vectors of size $O(10^4)$.



(a) Ritz values (b) Eigenvalues

Figure 7: Ritz values and approximate eigenvalues, i.e., reciprocal Ritz values, of the delay problem computed with variant T$\mathcal{FB}$. An eigenvalue is classified as converged if $E(\lambda, x) \leq 10^{-10}$.



(a) Iteration vs relative residual (b) Wall time vs relative residual

Figure 8: Comparison of C$\mathcal{B}$ and C$\mathcal{FB}$ for (5.2) with $n = 10.001$. The variant C$\mathcal{FB}$ converges in less iterations than variant C$\mathcal{B}$. Furthermore, the computation time differs several orders of magnitude.

6 Conclusions and outlook

In this paper we have presented a new procedure to compute solutions to a type of nonlinear eigenvalue problem with a particular low-rank structure. We have constructed the algorithm such that it is equivalent to the Arnoldi method on a (infinite dimensional) linear operator, and the behavior in the numerical examples is very similar to the Arnoldi method, including the restarting features. Although the construction is general, some specific adaptions, such as efficient formulas for y_0 in (3.5) and (3.12), are necessary in order to implement the algorithm for a specific problem. The numerical examples have illustrated that for large-scale problems the low-rank exploitation can result in significant lower computation times due to much lower orthogonalization and memory costs.

Several possible continuations of this result appear feasible. There are several variants of the Arnoldi method that might be extendible, e.g., a block Krylov-Schur [32], or advanced filtering techniques [6]. The understanding of the algorithm can also certainly be improved by further adapting results and understanding known for the standard Arnoldi method, e.g., [3, 15]. Due to the fact that the presented algorithm is equivalent to the Arnoldi method on an infinite-dimensional operator, convergence results will carry over under the condition that the properties in the proof are not properties of matrices.

Acknowledgment

This research was supported by the Dahlquist research fellowship, the Programme of Interuniversity Attraction Poles of the Belgian Federal Science Policy Office (IAP P6-DYSCO), by OPTEC, the Optimization in Engineering Center of the KU Leuven, by projects STRT1-09/33 and OT/10/038 of the KU Leuven Research Council,and by project G.0712.11N of the Research Foundation-Flanders (FWO).

References

[1] J. Asakura, T. Sakurai, H. Tadano, T. Ikegami, and K. Kimura, *A numerical method for nonlinear eigenvalue problems using contour integrals*, JSIAM Letters, 1 (2009), pp. 52–55.

[2] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. A. van der Vorst, *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*, SIAM, Philadelphia, 2000.

[3] M. Bellalij, Y. Saad, and H. Sadok, *Further analysis of the Arnoldi process for eigenvalue problems*, SIAM J. Numer. Anal., 48 (2010), pp. 393–407.

[4] T. Betcke, N. J. Higham, V. Mehrmann, C. Schröder, and F. Tisseur, *NLEVP: A collection of nonlinear eigenvalue problems*, ACM Trans. Math. Softw., 39 (2013), pp. 1–28.

[5] W.-J. Beyn, *An integral method for solving nonlinear eigenvalue problems*, Linear Algebra Appl., 436 (2012), pp. 3839–3863.

[6] Z. Bujanović and Z. Drmač, *A new framework for polynomial filtering in implicitly restarted Arnoldi type algorithms.* Extended abstract, Householder symposium XIX, June 8-13, 2014.

[7] C. Effenberger, *Robust solution methods for nonlinear eigenvalue problems*, PhD thesis, EPF Lausanne, 2013.

[8] C. Effenberger and D. Kressner, *Chebyshev interpolation for nonlinear eigenvalue problems*, BIT, 52 (2012), pp. 933–951.

[9] G. Golub and C. V. Loan, *Matrix computations*, The Johns Hopkins University Press, 2013. 4th edition.

[10] M. Gugat, *Boundary feedback stabilization by time delay for one-dimensional wave equations*, IMA J. Math. Control Inf., 27 (2010), pp. 189–203.

[11] S. Güttel, R. Van Beeumen, K. Meerbergen, and W. Michiels, *NLEIGS: A class of robust fully rational Krylov methods for nonlinear eigenvalue problems*, tech. rep., Department of Computer Science, KU Leuven, Leuven, 2013.

[12] E. Jarlebring, K. Meerbergen, and W. Michiels, *A Krylov method for the delay eigenvalue problem*, SIAM J. Sci. Comput., 32 (2010), pp. 3278–3300.

[13] ——, *Computing a partial Schur factorization of nonlinear eigenvalue problems using the infinite Arnoldi method*, SIAM J. Matrix Anal. Appl., 35 (2014), pp. 411–436.

[14] E. Jarlebring, W. Michiels, and K. Meerbergen, *A linear eigenvalue algorithm for the nonlinear eigenvalue problem*, Numer. Math., 122 (2012), pp. 169–195.

[15] Z. Jia, *The convergence of generalized Lanczos methods for large unsymmetric eigenproblems*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 843–862.

[16] D. Kressner, *A block Newton method for nonlinear eigenvalue problems*, Numer. Math., 114 (2009), pp. 355–372.

[17] R. Lehoucq and D. Sorensen, *Deflation techniques for an implicitly restarted Arnoldi iteration*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 789–821.

[18] V. Mehrmann and H. Voss, *Nonlinear eigenvalue problems: A challenge for modern eigenvalue methods*, GAMM-Mitt., 27 (2004), pp. 121–152.

[19] A. Neumaier, *Residual inverse iteration for the nonlinear eigenvalue problem*, SIAM J. Numer. Anal., 22 (1985), pp. 914–923.

[20] Y. Saad, *Numerical methods for large eigenvalue problems*, SIAM, 2011.

[21] G. L. Sleijpen, A. G. Booten, D. R. Fokkema, and H. A. van der Vorst, *Jacobi-Davidson type methods for generalized eigenproblems and polynomial eigenproblems*, BIT, 36 (1996), pp. 595–633.

[22] G. W. Stewart, *A Krylov–Schur algorithm for large eigenproblems*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 601–614.

[23] Y. Su and Z. Bai, *Solving rational eigenvalue problems via linearization*, SIAM J. Matrix Anal. Appl., 32 (2011), pp. 201–216.

[24] F. Tisseur and K. Meerbergen, *The quadratic eigenvalue problem*, SIAM Rev., 43 (2001), pp. 235–286.

[25] G. Unger, *Analysis of Boundary Element Methods for Laplacian Eigenvalue Problems*, 2010.

[26] R. Van Beeumen, K. Meerbergen, and W. Michiels, *A rational Krylov method based on Hermite interpolation for nonlinear eigenvalue problems*, SIAM J. Sci. Comput., 35 (2013), pp. A327–A350.

[27] H. Voss, *An Arnoldi method for nonlinear eigenvalue problems*, BIT, 44 (2004), pp. 387 – 401.

[28] ——, *A Jacobi-Davidson method for nonlinear and nonsymmetric eigenproblems*, Computers and structures, 85 (2007), pp. 1284–1292.

[29] ——, *Handbook of Linear Algebra, Second Edition*, no. 164 in Discrete Mathematics and Its Applications, Chapman and Hall/CRC, 2013, ch. Nonlinear Eigenvalue Problems.

[30] H. Voss, K. Yildiztekin, and X. Huang, *Nonlinear low rank modification of a symmetric eigenvalue problem*, SIAM J. Matrix Anal. Appl., 32 (2011), pp. 515–535.

[31] J. Wu, *Theory and Applications of Partial Functional Differential Equations*, Applied Mathematical Sciences. 119. New York, NY: Springer., 1996.

[32] Y. Zhou and Y. Saad, *Block Krylov-Schur method for large symmetric eigenvalue problems*, Numer. Algorithms, 47 (2008), pp. 341–359.